Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (currently amended) A method for processing data in a programmable processor, the method comprising:

decoding and executing instructions that instruct a computer system to perform operations,

at least some of the instructions including ~~a~~ group floating-point ~~instruction~~ instructions each operating on first and second registers partitioned into a plurality of floating point operands, the floating point operands having a defined precision and the defined precision being dynamically variable, having a defined result precision which is equal to the defined precision of the operands;

at least ~~some~~ one group floating-point instruction being a group floating-point multiply-and-add instruction, further operating on a third register partitioned into a plurality of floating-point operands,

operable to multiply the plurality of floating-point operands in the first and second registers and add the plurality of floating-point operands in the third register, each producing a floating-point value to provide a plurality of floating-point values, each of the floating-point values capable of being represented by the defined result precision, and a catenated result having a plurality of partitioned fields for ~~receiving~~ the plurality of floating point values.

2. (currently amended) The method of claim 1, wherein

at least ~~some~~ one group floating-point instruction being ~~at least one~~ a member of the collection consisting of group floating-point subtract, group floating-point add, and group floating-point multiply,

operable to perform a subtract, add ~~and~~ or multiply respectively on the plurality of floating-point operands in the first and second registers, ~~each producing a floating-point value~~ to provide a plurality of floating-point values, each of the floating-point values capable of being represented by the defined result precision, and a catenated result having a plurality of partitioned fields for ~~receiving~~ the plurality of floating point values; and

at least ~~some~~ one group floating-point instruction being ~~at least one~~ a member of the collection consisting of group floating-point set less, and group floating-point set greater or equal,

operable to perform a set-less ~~and~~ or set-greater-or-equal operation, respectively, on the plurality of floating-point operands in the first and second registers, ~~each producing a value~~ to provide a plurality of values, each of the values capable of being represented by the defined result precision, and a catenated result having a plurality of partitioned fields for ~~receiving~~ the plurality of values, wherein the value is zero if the operation produces a false result, and wherein the value is an identity value if the operation produces a true result; and

at least some of the instructions comprising performing data manipulations on multiple operands stored in partitioned fields of registers wherein the data manipulations comprise copying or rearranging operands.

3. (original) The method of claim 2 wherein the zero value and the identity value are values that construct a bit mask operable to select between alternate expressions using a bitwise Boolean operation.

4. (original) The method of claim 1 wherein the catenated result has a width of 128 bits.

5. (original) The method of claim 1 wherein the catenated result is provided to a register.

6. (original) The method of claim 1 wherein the defined precision is 16 bits.

7. (currently amended) The method of claim 1 wherein the defined precision is a format comprising one sign bit, five exponent bits and ten ~~significant~~ significand bits.

8. (original) The method of claim 1 wherein the defined precision is 32 bits.

9. (currently amended) The method of claim 1 wherein the precision of the group floating-point instructions is a format comprising one sign bit, eight exponent bits and 23 ~~significant~~ significand bits.

10. (original) The method of claim 1 wherein the defined precision is 64 bits.

11. (currently amended) The method of claim 1 wherein the precision of the group floating-point instructions is a format comprising one sign bit, eleven exponent bits and 52 ~~significant~~ significand bits.

12. (currently amended) A computer-readable storage medium having stored therein a plurality of instructions that cause a computer processor to perform data operations:

at least some of the instructions including ~~a~~ group floating-point instructions each operating on first and second registers partitioned into a plurality of floating point operands, the floating point operands having a defined precision and the defined precision being dynamically variable, having a defined result precision which is equal to the defined precision of the operands;

the group floating-point instructions including a group floating-point multiply-and-add instruction, further operating on a third register partitioned into a plurality of floating-point operands,

the group floating-point multiply-and-add instruction operable to multiply the plurality of floating-point operands in the first and second registers and add the plurality of floating-point operands in the third register, each producing a floating-point value to provide a plurality of floating-point values, each of the floating-point values capable of being represented by the defined result precision, and a catenated result having a plurality of partitioned fields for ~~receiving~~ the plurality of floating point values.

13. (currently amended) The computer-readable storage medium of claim 12,

at least ~~some~~ one group floating-point instruction being ~~at least one~~ a member of the collection consisting of group floating-point subtract, group floating-point add, and group floating-point multiply,

operable to perform a subtract, add ~~and~~ or multiply respectively on the plurality of floating-point operands in the first and second registers, ~~each producing a floating-point value~~ to provide a plurality of floating-point values, each of the floating-point values capable of being represented by the defined result precision, and a catenated result having a plurality of partitioned fields for ~~receiving~~ the plurality of floating point values; and

at least ~~some~~ one group floating-point instruction being ~~at least one~~ a member of the collection consisting of group floating-point set less, and group floating-point set greater or equal,

operable to perform a set-less ~~and~~ or set-greater-or-equal operation, respectively, on the plurality of floating-point operands in the first and second registers, ~~each producing a value~~ to provide a plurality of values, each of the values capable of being represented by the defined result precision, and a catenated result having a plurality of partitioned fields for ~~receiving~~ the plurality of values, wherein the value is zero if the operation produces a false result, and wherein the value is an identity value if the operation produces a true result; and

at least some of the instructions comprising performing data manipulations on multiple operands stored in partitioned fields of registers wherein the data manipulations comprise copying or rearranging operands.

14. (previously presented) The computer-readable storage medium of claim 13 wherein the zero value and the identity value are values that construct a bit mask operable to select between alternate expressions using a bitwise Boolean operation.

15. (previously presented) The computer-readable storage medium of claim 12 wherein the catenated result has a width of 128 bits.

16. (previously presented) The computer-readable storage medium of claim 12 wherein the catenated result is provided to a register

17. (previously presented) The computer-readable storage medium of claim 12 wherein the defined precision is 16 bits.

18. (currently amended) The computer-readable storage medium of claim 12 wherein the defined precision is a format comprising one sign bit, five exponent bits and ten ~~significant~~ significand bits.

19. (previously presented) The computer-readable storage medium of claim 12 wherein the defined precision is 32 bits.

20. (currently amended) The computer-readable storage medium of claim 12 wherein the precision of the group floating-point instructions is a format comprising one sign bit, eight exponent bits and 23 ~~significant~~ significand bits.

21. (previously presented) The computer-readable storage medium of claim 12 wherein the defined precision is 64 bits.

22. (currently amended) The computer-readable storage medium of claim 12 wherein the precision of the group floating-point instructions is a format comprising one sign bit, eleven exponent bits and 52 ~~significant~~ significand bits.

23. - 32. (canceled)

33. (previously presented) A method for performing data operations in a programmable processor comprising:

executing a plurality of instructions each of which (i) operates on data stored in a first, a second and a third register, the data in the first register comprising a first plurality of equal-sized data elements, the data in the second register comprising a second plurality of equal-sized data elements, the data in the third register comprising a third plurality of equal-sized data elements, (ii) multiplies each data element in the first register with a corresponding data element in the second register to produce a plurality of products, and (iii) adds each product in the plurality of products to a corresponding data element in the third register to produce a plurality of sums, and (iv) provides the plurality of sums as a catenated result;

wherein the plurality of instructions includes a floating-point instruction that operates on floating-point data elements stored in the first, second and third registers.

34. (previously presented) The method of claim 33 wherein each of the plurality of instructions includes a field that indicates the size of each of the first plurality and second plurality of data elements.

35. (previously presented) The method of claim 33 wherein the catenated result is returned to a fourth register.

36. (previously presented) The method of claim 33 wherein for the floating-point instruction, each of the first plurality and second plurality of equal-sized data elements is a floating-point value that is n bits wide, and each of the third plurality of equal-sized data elements is also a floating-point value that is n bits wide.

37. (previously presented) The method of claim 36 wherein the floating-point instruction multiplies data elements of 32-bit floating-point data and adds data elements of 32-bit floating-point data.

38. (previously presented) The method of claim 33 wherein the plurality of instructions includes an integer instruction that operates on integer data elements stored in the first, second and third registers.

39. (previously presented) The method of claim 38 wherein for the integer instruction, each of the first plurality and second plurality of equal-sized data elements is an integer value that is n bits wide, and each of the third plurality of equal-sized data elements is an integer value that is 2*n bits wide.

40. (previously presented) The method of claim 39 wherein the integer instruction multiplies data elements of 8-bit integer data and adds data elements of 16-bit integer data.

41. (previously presented) The method of claim 39 wherein the integer instruction multiplies data elements of 16-bit integer data and adds data elements of 32-bit integer data.

42. (previously presented) The method of claim 39 wherein the integer instruction multiplies data elements of 32-bit integer data and adds data elements of 64-bit integer data.

43. (previously presented) A computer-readable storage medium having stored therein instructions that cause a computer processor to perform operations on data stored in registers in the computer processor, the instructions comprising:

a plurality of instructions each of which (i) operates on data stored in a first, a second and a third register, the data in the first register comprising a first plurality of equal-sized data elements, the data in the second register comprising a second plurality of equal-sized data elements, the data in the third register comprising a third plurality of equal-sized data elements, (ii) multiplies each data element in the first register with a corresponding data element in the second register to produce a plurality of products, and (iii) adds each product in the plurality of products to a corresponding data element in the third register to produce a plurality of sums, and (iv) provides the plurality of sums as a catenated result;

wherein the plurality of instructions includes a floating-point instruction that operates on floating-point data elements stored in the first, second and third registers.

44. (previously presented) The computer-readable storage medium of claim 43 wherein each of the plurality of instructions includes a field that indicates the size of each of the first plurality and second plurality of data elements.

45. (previously presented) The computer-readable storage medium of claim 43 wherein the catenated result is returned to a fourth register.

46. (previously presented) The computer-readable storage medium of claim 43 wherein for the floating-point instruction, each of the first plurality and second plurality of equal-sized data elements is a floating-point value that is n bits wide, and each of the third plurality of equal-sized data elements is also a floating-point value that is n bits wide.

47. (previously presented) The computer-readable storage medium of claim 46 wherein the floating-point instruction multiplies data elements of 32-bit floating-point data and adds data elements of 32-bit floating-point data.

48. (previously presented) The computer-readable storage medium of claim 43 wherein the plurality of instructions includes an integer instruction that operates on integer data elements stored in the first, second and third registers.

49. (previously presented) The computer-readable storage medium of claim 48 wherein for the integer instruction, each of the first plurality and second plurality of equal-sized data elements is an integer value that is n bits wide, and each of the third plurality of equal-sized data elements is an integer value that is 2*n bits wide.

50. (previously presented) The computer-readable storage medium of claim 49 wherein the integer instruction multiplies data elements of 8-bit integer data and adds data elements of 16-bit integer data.

51. (previously presented) The computer-readable storage medium of claim 49 wherein the integer instruction multiplies data elements of 16-bit integer data and adds data elements of 32-bit integer data.

52. (previously presented) The computer-readable storage medium of claim 49 wherein the integer instruction multiplies data elements of 32-bit integer data and adds data elements of 64-bit integer data.